Scientia Iranica, Vol. 15, No. 1, pp 75 93

SCIENTIA IRANICA

Modeling and Intelligent Control of a Robotic Gas Metal Arc Welding System

H. Sayyaadi* **and A.A. Eftekharian**[1]

Welding is an important manufacturing process that can be automated and optimized. This paper focuses on the development of a robotic arc welding system, wherein a three-degree-of-freedom Selective Compliance Assembly Robot Arm (SCARA) is interfaced to a Gas Metal Arc Welding (GMAW) process. The entire system is composed of a robot linked to a GMAW system. Set points are derived using the desired mass and heat input, along with the weld speed. The stick-out and the current of the welding process are controlled using an Adaptive Neural Network Controller (ANNC). The trajectory of the robot or the weld profile is also controlled by implementing a Mixed Fuzzy-GA Controller (MFGAC) on a three-axis SCARA robot. The system is, then, analyzed and the results show adequate improvement in the efficiency and performance of the proposed controller in welding a curvilinear weld profile.

INTRODUCTION

Being the most popular process, gas metal arc welding is an important component in many industrial and manufacturing operations and high quality welding procedures are essential to overall product quality in any production facility.

The practical industrial welding task is difficult. In order to make high quality weld, it is required to implement experienced weld operators. One reason for this is the need to have the welding parameters properly adjusted for a given task, in order to get good results. Generally, a good weld is identified by its microstructure and by various factors (e.g., the amount of spatter, or the amount of overfill or underfill, etc.). Although these are not easily measured or quantified, they can be related to characteristics, such as the cooling rate of the weld pool, the metal transfer mode, the bead/groove geometry and work piece defects etc. Likewise, many of these characteristics can be related to the mass and heat transferred from the GMAW process to the weld pool.

The specific focus of this paper is the automation of the GMAW process by the combination of a robot integrated gas metal arc welding system and by the use of intelligent controllers, such as neural networks and fuzzy method. The use of control in the GMAW process can eliminate much of the "guess work" often used by welding engineers to specify welding parameters for a given task.

Advanced and intelligent methods for controlling the welding process can lead to significant improvements in the economic competitiveness of an industry [1].

Smartt and Einerson [2] developed a steady-state model for heat distribution, based on the unit length of the weld and mass (the transverse cross-sectional area of the deposited metal) transferred from the electrode to the work piece in the GMAW process. Using the relation for heat and mass, a PI-based control system was developed for maintaining the desired heat and mass by regulating the current.

Abdel Rahman [3] developed a model-based nonlinear controller, which uses nonlinear state feedback to exactly linearize and decouple the GMAW system. The linearized system is then controlled using 2-PI controllers. In [3], the design of the controller is based on the simplified model for the GMAW process.

Jalili et al. [4] developed a feedback linearization controller based on a sliding mode control action for a GMAW system. The controller uses a nonlinear state feedback to exactly linearize and decouple the process. The linearized system is then controlled using a model reference sliding mode controller.

Successful applications of a pseudo-gradient adaptive algorithm, for self-tuning a PI-based puddle width

*. *Corresponding Author, Department of Mechanical Engineering, Sharif University of Technology, Tehran, I.R. Iran.*

1. *Department of Mechanical Engineering, Sharif University of Technology, Tehran, I.R. Iran.*

controller, for a consumable electrode of the GMAW process, were given by Henderson et al. [5].

Doumanidis [6] developed an adaptive, Multi-Input Multi-Output (MIMO) scheme to control both geometrical and thermal characteristics of a weld, based on lumped parameters and distributed parameters in modeling and identification.

The problem of the adaptive and decoupling control of the MIMO welding process was addressed by Cook et al. [7]. Moore, Yender and Naidu [8] developed the analytical and experimental model of the GMAW and, then, used single-output and multi-loop PI control strategies to control the process. As discussed in [7], the variables that are close to the process in the control algorithm were used.

Golob et al. [9] proposed a fuzzy controller for the GMAW process; they modeled the self-regulation process analytically, developed various dynamic models and, then, proposed a welding current control, based on a fuzzy PI controller.

Bingal and Cook [10] developed a model to correlate the anode temperature profile with the dynamic melting rate in a gas metal arc welding system. Components of this model are identified as the electrode-melting rate and the temperature-dependent resistivity of the electrode and arc voltage. The differential equations describing the dynamic behavior of the electrode extension were derived from mass continuity and energy relations and, then, a final model was developed for the GMAW dynamics.

Here, a simplified second order nonlinear GMAW model and the models for heat and mass transfer are used to independently control current and arc length, using the open circuit voltage and wire feed speed. Since the mass and heat transferred to the work piece determine the quality of the welding, it is necessary to achieve the desired mass and heat values for a particular welding. Thus, assuming that the contact tip to work piece distance and weld torch speed are being held constant during welding, one approach to controlling the quality of the weld in the GMAW process is to control the current and stick-out, which could define the arc voltage, so as to control the heat and mass input to the weld pool. Therefore, this paper presents the design of an adaptive neural network controller for the GMAW process. Attractive features of this type of adaptive controller include the lack of dependency on the process parameters, the robust behavior of the controller to noise, disturbance and uncertainty and, finally, ease of implementation. In the robotic part of the system in this work, a mixed fuzzy GA controller is presented for controlling a 3-axis SCARA robot.

The main difference between MIMO control systems and Single-Input Single-Output (SISO) control systems is in the means of estimating and compensating for the interaction between the degrees of freedom. MIMO systems usually posses complicated dynamic coupling. Thus, estimating the accurate dynamic model and decoupling it for designing the controller is difficult. Hence, the traditional model-based SISO control scheme is hard to implement on complicated MIMO systems, as the computational procedure is large. Therefore, model-free intelligent control strategies are gradually attracting attention. Although the fuzzy control theory has been successfully employed in many control engineering fields [11-15], its control strategies were mostly designed for SISO systems, in spite of the effect of dynamic coupling on MIMO systems. Additionally, the number of control rules and controller computational burdens grows larger, exponentially, with the number of variables considered. Clearly, the difficulty in controlling MIMO systems is how to solve the coupling effects between the degrees of freedom. Therefore, an appropriate coupling fuzzy controller is incorporated into a Traditional Fuzzy Controller (TFC) for controlling a MIMO robotic system in order to compensate for the dynamic coupling effects between the degrees of freedom.

After constructing the mixed fuzzy controller, the use of genetic algorithms in the design and implementation of the mixed fuzzy logic controller is investigated. Previously, a generation of membership functions had been a task mainly undertaken either iteratively, by trial-and-error, or by human experts. A task such as this is a natural candidate for the GA, since GA's attempt is to create membership functions that will cause the controller to perform optimally. Recently, much work has been done using GA to do this task. Karr, for example, used a GA to generate membership functions for a PH control process [16] and a cart-pole problem [17]. Such work has shown GA's ability to successfully create the individual parts of a fuzzy controller. Mohammadian and Stonier [18] developed a fuzzy logic controller and optimized the membership functions using genetic algorithms. Mester, in [19], developed a neuro-fuzzy-genetic controller for robot manipulators; he applied the genetic algorithm for the fuzzy rules set optimization. Eskil, Efe and Kaynak, in [20], proposed a procedure for T-Norm adaptation in fuzzy logic systems, using genetic algorithms. They investigated the performance of a fuzzy system, having parameterized T-Norm in control of the robot manipulators.

In this paper, first, a mixed fuzzy control system is constructed and, then, a genetic algorithm is applied to tune and optimize the membership functions and scaling factors in such a manner that a high precision controller for the trajectory tracking of a SCARA robot is obtained. This mixed fuzzy-GA controller can effectively remove the coupling effects of the robotic system and the control strategy proposed in this paper

is very easy to design and implement. This paper is organized as follows: First, some reviews about the modeling of a GMAW process are presented and, then, the control objective is discussed. Subsequently, an adaptive neural network controller is designed. Applications of the adaptive neural network control to the GMAW process and simulation results are given in the other section. Next, in the robotic part of the system, a mixed fuzzy control is designed and a genetic algorithm is introduced and implemented in the fuzzy controller for optimization. The simulation results of a curvilinear weld profile for robotic welding is, then, presented and, finally, the results are discussed and conclusions are drawn.

MODELING OF THE GMAW PROCESS

In this section, The GMAW system is briefly discussed. The power supply consists of a constant voltage source connected to the electrode and the work piece. The wire speed, S, travel speed of the torch, R, open-circuit voltage, V_{oc}, and contact tip to work piece distance, CT, are adjusted to get the desired weld. Here, x is the distance of the center of the mass of the droplet from the electrode [1]. Modeling of the GMAW process dynamics produces a fifth-order nonlinear differential equation. The model used in this work is the fourth-generation of the derivative equation that originated at the Idaho National Engineering and Environmental Laboratory (INEEL) and which was subsequently developed by INEEL, as well as Idaho State University (ISU) researchers and others [1,21]. Here, the results are not described in detail, but interested readers may refer to the following earlier work in GMAW modeling [10,22]. The state-space representations of the resulting equation are given in the following equations. The state variables are defined as follows:

$x_1 = x$ droplet displacement (m)
$x_2 = \dot{x}$ droplet velocity (m/sec)
$x_3 = m_d$ droplet mass (kg)
$x_4 = l_s$ stick-out (m)
$x_5 = I$ current (A)

The nonlinear state equations become as follows:

$$\dot{x}_1 = x_2,$$

$$\dot{x}_2 = \frac{kx_1 \quad Bx_2 + F_{tot}}{x_3},$$

$$\dot{x}_3 = M_R \rho_w,$$

$$\dot{x}_4 = u_1 \quad \frac{M_R}{\pi r_w^2},$$

$$\dot{x}_5 = \frac{u_2 \quad (R_a + R_s + R_L)x_5 \quad V_0 \quad E_a(CT \quad x_4)}{L_s}, \tag{1}$$

where $F_{tot} = F_{em} + F_d + F_m + F_g$ force acting on the droplet. The forces, F_{em}, F_d, F_m, F_g, are the electrode magnetic force, aerodynamic drag force, momentum force, and gravity force, respectively. k and B are the stiffness constant and damping coefficient of the droplet. R_a and R_s are the arc resistance and source resistance, respectively. ρ_w is the electrode density, r_w is the electrode radius, V_0 is the arc voltage constant, E_a is the arc length factor, L_s is the source inductance and M_R is the melting rate. The melting rate, M_R, and the electrode resistance, R_L, are given by:

$$M_R = C_2 \rho x_4 x_5^2 + C_1 x_5,$$

$$R_L = \rho \left[x_4 + \frac{1}{2} \left(\left(\frac{3x_3}{4\pi\rho x} \right)^{\frac{1}{3}} + x_1 \right) \right]. \tag{2}$$

The output equations can be as follows:

$$y_1 = V_0 + R_a x_5 + E_a(CT \quad x_4),$$

$$y_2 = x_5, \tag{3}$$

where the output variables are:

$y_1 = V_{arc}$ arc voltage (V),
$y_2 = I$ current (A),

and the control variables are:

$u_1 = S$ wire feed speed (m/sec),
$u_2 = V_{oc}$ open-circuit voltage (V).

The detachment of a droplet will occur, either due to the imbalance of forces on the droplet, which means:

$$F_{tot} > F_s. \tag{4}$$

Or due to the shape instability as follows:

$$r_d > \frac{\pi(r_d + r_w)}{1.25 \left(\frac{x + r_d}{r_d} \right) \left(1 + \frac{\mu_0 I^2}{2\pi^2 \gamma (r_d + r_w)} \right)^{\frac{1}{2}}}, \tag{5}$$

where F_s is the surface tension force, r_d is the droplet radius, μ_0 is the permeability of free space and γ is the surface tension of liquid steel [1]. Further equations of the droplet velocity and detachments criteria can be found in [1,10,22]. Figure 1 shows a schematic diagram of the GMAW process.

The GMAW dynamics model, given by Equations 1, is highly nonlinear, which makes the simulation or control of the process very difficult. To simplify the model, based on some approximations, first, consider

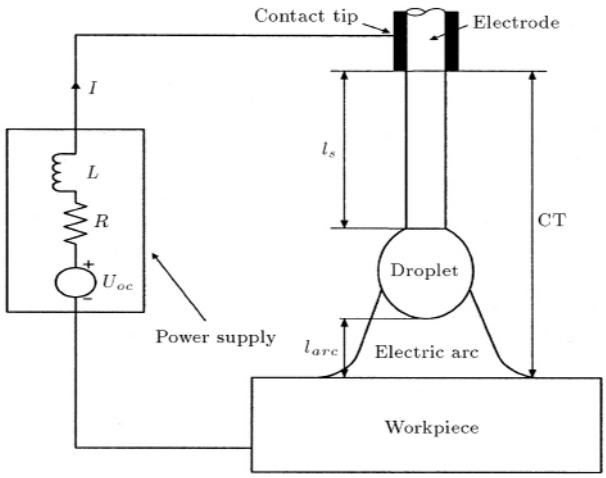


Figure 1. Schematic diagram of GMAW system.

the current, I, and stick-out, l_s, relations from Equations 1 as follows:

$$\dot{x}_4 = u_1 \quad \frac{M_r}{\pi r_w^2},$$

$$\dot{x}_5 = \frac{u_2 \quad (R_a + R_s + R_L)x_5 \quad V_0 \quad E_a(CT \quad x_4)}{L_s}. \tag{6}$$

In [1], it has been shown that:

$$R_L = \rho x_1. \tag{7}$$

So, the simplified model is:

$$\dot{x}_4 = u_1 \quad \left(\frac{C_2\rho}{\pi r_w^2}x_4x_5^2 + \frac{C_1}{\pi r_w^2}x_5\right),$$

$$\dot{x}_5 = \frac{u_2 \quad (R_a + R_s + \rho x_4)x_5 \quad V_0 \quad E_a(CT \quad x_4)}{L_s}. \tag{8}$$

The objective is to achieve the desired weld bead cross section, G_d [mm^2], and heat flow to the weld pool, H_d [J/mm]. This is achieved by considering the following relation:

$$G_d = \frac{M_R}{R} \times 10^9, \tag{9}$$

where R is the weld torch travel rate, [mm/s], and M_R is the melting rate, [m^3/s], given by [3]:

$$M_R = C_1 I + C_2 \rho I^2 l_s, \tag{10}$$

where C_1 and C_2 are constants, ρ [ohm/m] is the welding wire resistivity and l_s [m] is the stick-out of the welding wire.

$$H_d = \frac{EI\eta}{R} = \frac{I(V_e + V_{arc})\eta}{R}, \tag{11}$$

where I is the current from the electric source, V_e is the voltage drop across the electrode, V_{arc} is the voltage drop across the arc, E is the secondary circuit voltage drop, η is the heat transfer efficiency from the process to the base metal and R is the weld torch travel speed. In [4] V_{arc} and V_e are expressed as follows:

$$V_e = I(l_s + 0.5(x + r_d)),$$

$$V_{arc} = V_0 + E_a(CT \quad l_s) + IR_a, \tag{12}$$

where V_0, E_a and R_a are empirical constants and r_d and x are droplet radius and displacement, respectively. Since the droplet radius and displacement are hard to measure in a practical welding environment and are much smaller than the stick-out, they can be neglected in the calculation of the desired arc length to produce a certain heat input. Thus, given a desired weld bead cross section, G_d, and heat, H_d, Equations 9 through 12 can be used to find the desired current and stick-out, which can achieve the desired values of heat and mass transfer to the weldment [1,4]. These values of current and stick-out are then used as set points for the dynamic system in Equations 8 and an adaptive neural network controller is used to track the desired set points. It should be emphasized that in this scheme it is not required to control the torch travel speed, R, which can be adjusted through an outer loop control of thermal and fill processes.

ADAPTIVE NEURAL NETWORK CONTROLLER

Neural networks are becoming more and more popular and their use is growing day by day. This is because neural networks have many attractive features, such as learning ability, approximation of functions, robustness, adaptiveness and so on, which are capable of solving the problems with uncertainty factors. Therefore, in this research, an adaptive neural network structure is used to control the uncertain structure of the GMAW model and, also, to interact with different environmental disturbance and noise. Intense research has been undertaken to use neural networks as controllers. The control learning schemes, which are used in this paper, are discussed here. Control systems have influence on different features of modern life. A control system is, in the broadest sense, any interconnection of components to provide a desired function. The portion of a system to be controlled is called the plant or process. A controller is used to produce the desired behavior of a plant.

To train a neural network controller, supervised learning can be used. In order to do this, the plant's output is presented to traditional feedback linearization and the neural network controllers (Figure 2). The

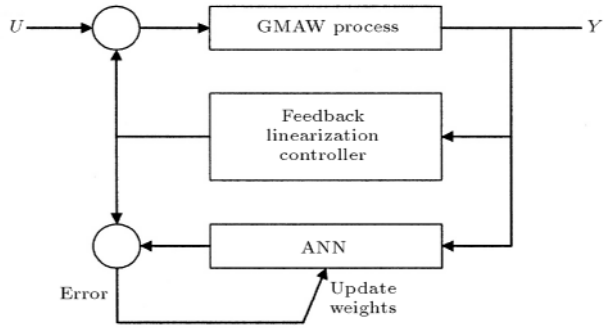


Figure 2. Supervised learning using an existing controller.

feedback linearization controller's output is used to control the plant and the plant's output is used as input to the neural network and feedback linearization controller. So, the neural network will be trained, as time goes by and will eventually become a controller. After completion of training, the feedback linearization controller can be ignored and the neural network will take control. However, this time another scheme, which is called unsupervised learning, is used (Figure 3). This kind of learning is very practical in cases where disturbances and uncertainties exist, because in this manner, the neural network can adapt its structure to the new conditions and can, therefore, perfectly maintain the system in a stable phase.

For control purposes, in this paper a feed forward controller is used with the structure 4-10-2. This means that a 3-layer network, with 4 neurons in the input layer, 10 neurons in the hidden layer and 2 neurons in the output layer, is used. The choice of hidden layer size is one of the most important considerations for neural network design and this area of study is still under intensive research, with no conclusive solutions yet available. The exact analysis of this issue is quite difficult, due to the complexity of the network's mapping and the nondeterministic nature of many successfully completed training procedures. In this work, the number of neurons in the hidden layer is determined by the trial and error approach. Several attempts have been made to study network performance with different numbers of neurons. The number of neurons within the hidden layer is selected, based on the accuracy of the control process.

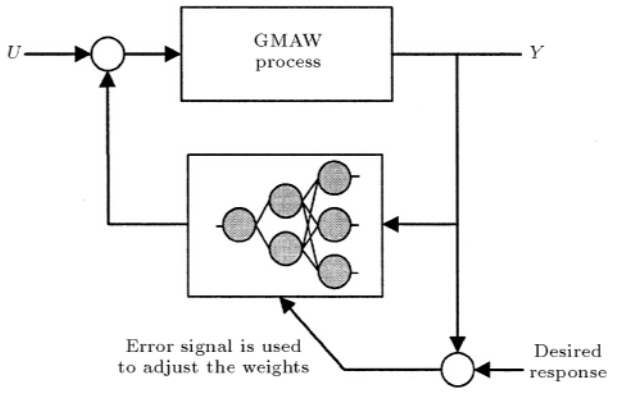


Figure 3. Unsupervised learning adaptive neural network.

The structure of the network used here is a 3-layer perceptron with an extended back propagation algorithm; the activation function, which in the neuron units is a general kind of sigmoid function:

$$f(x,U,L,T)=\frac{U\quad L}{1+e^{\frac{x}{T}}}+L, \tag{13}$$

where U, L and T are the upper bound, lower bound and slope of the activation function, respectively. The function is reduced to the usual sigmoid when $L=0$, $U=1$ and $T=1$. The Enhanced Back-Propagation Algorithm (EBPA) updates not only the matrix of the weights between the input/hidden layers and hidden/output layers, $W(K)$ and $V(K)$, respectively, but also the parameters, U, L and T, of the neurons at the hidden and output layers [23,24]. In [25], it is shown that the EBPA algorithm performs equally as well as a simple BP, in terms of learning speed and approximation accuracy. These characteristics are particularly suitable for online learning problems. Here, the EBPA, which is used in the learning procedure, is explained.

Knowing that $e(t)=y(t)\quad y_s(t)$, the error vector at time t, where $y(t)$ stands for plant output and $y_s(t)$ stands for network output, is defined by the following equations. At any given time, t, if $x(t)$ is the input vector, then, the ith element of the output vector at the hidden layer is as follows:

$$z(i,t)=L_h(i,t)+\frac{U_h(i,t)\quad L_h(i,t)}{1+e^{\frac{(V(i,t)\times x(t)+P_V(i,t))}{T_h(i,t)}}}, \tag{14}$$

where $L_h(i,t)$ is the ith element of the lower limit vector (for the hidden layer), $U_h(i,t)$ is the ith element of the upper limit vector (for the hidden layer), $T_h(i,t)$ is the ith element of the slope vector (for the hidden layer), $V(i,t)$ is the ith row of the (input to hidden layer) weight matrix and $P_V(i,t)$ is the ith element of the (input to hidden layer) threshold vector. The ith element of the output vector at the network's output is as follows:

$$Y_s(i,t)=L_O(i,t)+\frac{U_O(i,t)\quad L_O(i,t)}{1+e^{\frac{(W(i,t)\times z(t)+P_W(i,t))}{T_O(i,t)}}}, \tag{15}$$

where $L_O(i,t)$ is the ith element of the lower limit vector (for the output layer), $U_O(i,t)$ is the ith element of the upper limit vector (for the output layer), $T_O(i,t)$ is the ith element of the slope vector (for the output layer), $W(i,t)$ is the ith row of the (hidden to output layer) weight matrix and $P_W(i,t)$ is the ith element of the (hidden to output layer) threshold vector. Vector

$X(t)$, containing the first $n_h \times (n_i + n_o) + 4 \times (n_o + n_h)$ neural network states, where n_i, n_h and n_o are the number of neurons in input, hidden and output layers, respectively, is updated according to an extended gradient rule as follows:

$$X(t+T) = X(t) \quad \eta \times \frac{\partial y_s}{\partial X} \times e(t) + Z(t), \tag{16}$$

where η is the learning rate, $\frac{\partial y_s}{\partial X}$ is the Jacobian matrix, T is the sampling time and $Z(t)$ represents an additional contribution from a filtered error as follows:

$$D(t+1) = \alpha \times D(t) \quad \eta \times \frac{\partial y_s}{\partial X} \times e(t),$$

$$Z(t) = \alpha \times D(t). \tag{17}$$

The filter decay rate, α, is called "momentum", if $\alpha = 0$, the update law reduces to the classic gradient rule. It can be seen that $Z(t)$ somehow represents all past increments of $X(t)$. The final part of the whole state vector is $D(t)$.

The three most important parameters η, α and T must be chosen before the learning procedure. The effectiveness and convergence of the back-propagation learning algorithm depends significantly on the value of these learning constants, which is strongly related to the class of the learning problem and the network architecture. In general, the optimal value of the learning constants will be decided only for the given problem and there is not a unique value suitable for different training cases. Therefore, the values of the learning constants have to be chosen experimentally by a trial and error approach. In this study, the development and training of the network were carried out by using the MATLAB neural network toolbox, the SIMULINK toolbox [26] and, also, the neural network toolbox [25], available on the Mathworks website [27].

APPLYING ADAPTIVE NEURAL NETWORK CONTROLLER TO GMAW PROCESS

The first step in the design of a controller for the GMAW process is designing a conventional controller, in order to learn the neural network controller. For this task, a feedback linearization controller is used. The first step is to choose a nonlinear state feedback to obtain a linearized system. Then, a simple controller for the linearized system is designed. The nonlinear state feedback can be obtained by an inspection of the system in Equations 8. The following control law for current control is chosen:

$$u_2 = \rho x_4 x_5 + V_0 + E_a(CT \quad x_4) + L_s V_2, \tag{18}$$

which results in the linearized current equation as follows:

$$\dot{x}_5 = \frac{R_a + R_s}{L_s} x_5 + V_2. \tag{19}$$

The control law for the stick-out loop is as follows:

$$u_1 = \frac{C_1 s_5 + C_2 \rho x_5^2 x_4}{\pi r_w^2} + V_1, \tag{20}$$

which results in the following linearized stick-out equation:

$$\dot{x}_4 = V_1. \tag{21}$$

Thus, the linearized system is obtained and current and stick-out dynamics are decoupled. A controller that can achieve a set point tracking could be as simple as a PI controller, for both the current loop and a stick-out loop [3]. Hence:

$$V_1 = K_{p1}(x_{4d} \quad x_4) + K_{i2} \int_0^t (x_{4d} \quad x_4)dt,$$

$$V_2 = K_{p2}(x_{5d} \quad x_5) + K_{i2} \int_0^t (x_{5d} \quad x_5)dt, \tag{22}$$

where x_{4d} and x_{5d} are the desired values for the current and stick-out, respectively, and K_{p1}, K_{i1}, K_{p2} and K_{i2} are the controller's gains. Figure 4 shows a schematic diagram of the nonlinear controller.

The most important parameter values, which are considered in the GMAW process, are presented in Table 1 (interested readers are referred to [21] for a complete list of welding parameters). For training the neural network controller from an existing controller, 4 inputs and 2 outputs of the classic controller are used as the input and output signals for the neural network. The four inputs are reference values of the current and the stick-out, which can be attained through Equations 9 to 12, and the feedback values of the current and the stick-out, which can be attained from the feedback loop of the classic controller. The two outputs are S and V_{oc}, which are used as signals to make the output error and train the network (Figure 5). After training is completed, the neural network controller is substituted

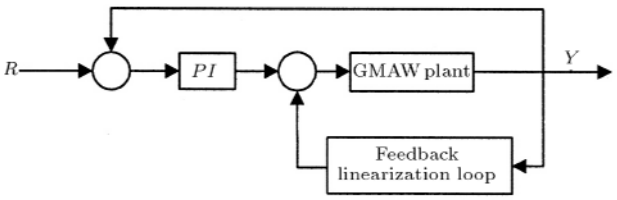


Figure 4. Schematic diagram of nonlinear controller.

Table 1. The system parameters.

Parameter	Value
C_1	2.8855e-10 m^3/A
C_2	5.22e-10 $m^3/A^2.Ohm$
ρ	0.2821 Ohm/m
E_a	1500 V/m
V_a	15.7 V
R_a	0.022 Ohm
R_S	0.001 Ohm
L_s	0.35 mH
R	8 m/s
CT	0.01 m
r_w	0.001 m

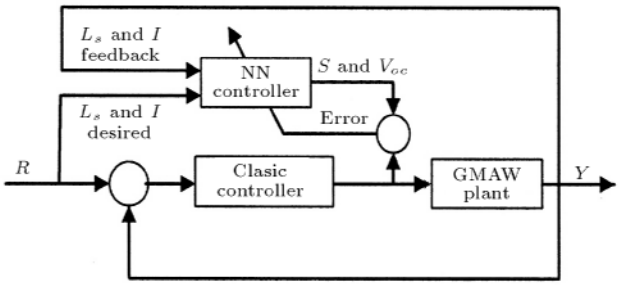


Figure 5. Schematic diagram of the training process.

for the classic controller (Figure 3). In this manner, the neural network controller can deal perfectly with both tracking and disturbance rejection tasks. In order to evaluate the algorithm, five different cases, as shown in Table 2, were considered for the simulation. The results of the simulation are shown in Figures 6 through 9.

As can be seen, Figure 6a shows the tracking response of the proposed controller for the stick-out and Figure 6b shows the tracking response of the current for Case 1 in Table 2. The simulation results of other cases are shown in Figure 7. Figure 8 shows the response of the neural network controller, in cases where 50% uncertainty exists in the parameters of the GMAW model. The parameters considered for uncertainty are: E_a, ρ, R_a, R_s, L_s and R, which are weld parameters identified formerly. The 50% model

Table 2. Cases considered for simulation.

Cases	G_d	H_d	I_d	L_{sd}
Case 1	1.5e-5	5e5	215.8561	0.0143
Case 2	1.5e-5	6.5e5	235.0264	0.0106
Case 3	1.5e-5	7e5	241.6517	0.0096
Case 4	2.05e-5	5.5e5	196.7447	0.0174
Case 5	2.05e-5	6e5	200.8751	0.0163

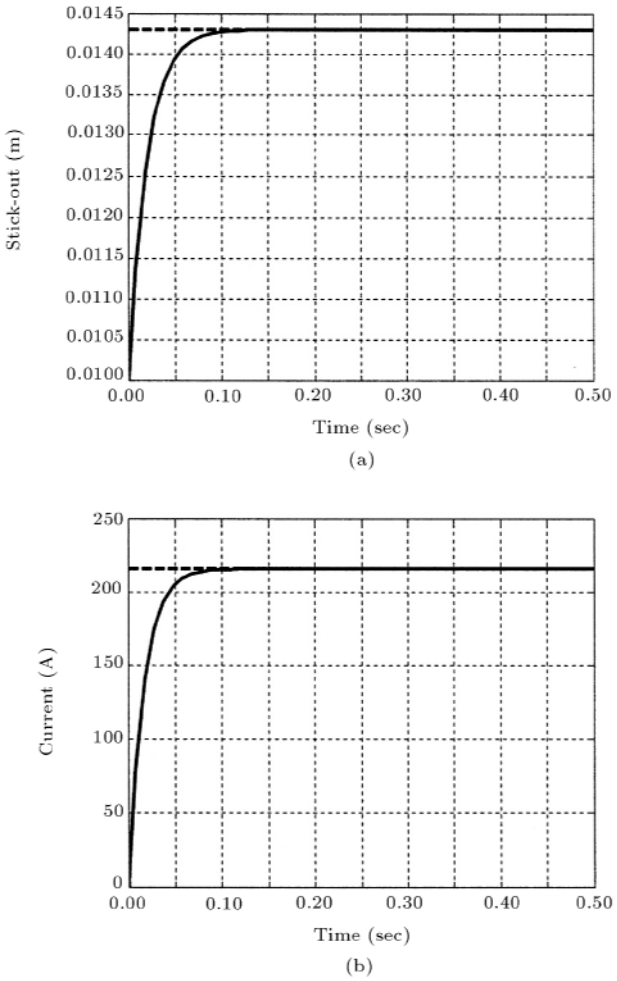


Figure 6. Desired and actual responses for stick-out (a) and current (b) for Case 1 of neural network controller.

uncertainty is applied to these six parameters with 50% tolerance as $(x \pm 50\%x)$. When these uncertainties are applied to the GMAW model, considerable differences between this new case and the previous one will occur. The new case may be considered as an actual system that differs from the old case, which represents the ideal model. Thus, the controller, as illustrated in Figure 8, has to overcome these changes and satisfy the stable performance of the system. As illustrated in the figure, the neural network becomes stable after 0.2 sec and overcomes the changes. The uncertainties applied for evaluating the performance of the controller are derived from the actual welding process, which encounters different sources of noise and disturbance, such as wind blow, temperature variations, air dampness and additional vibrations. As stated in [7,8,21], these disturbances are considered maximum up to 30% changes in the parameters, but in this research, 50% changes are applied to these parameters in order to verify the adaptation effectiveness of the proposed controller. The neural network weights are drawn in Figure 9. From this figure, it is clear that the

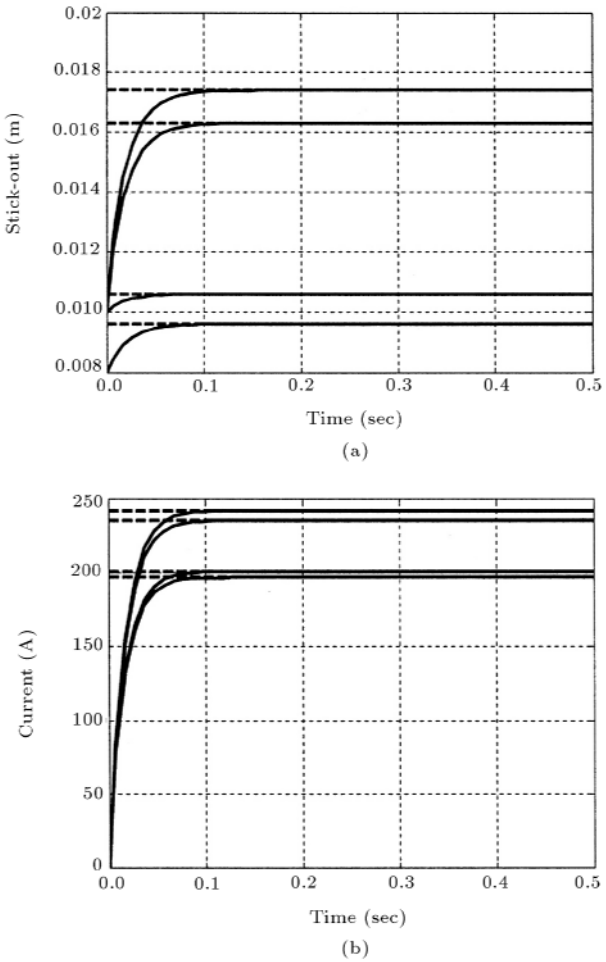


Figure 7. Desired and actual responses for stick-out (a) and current (b) for Cases 2 to 5 of neural network controller.

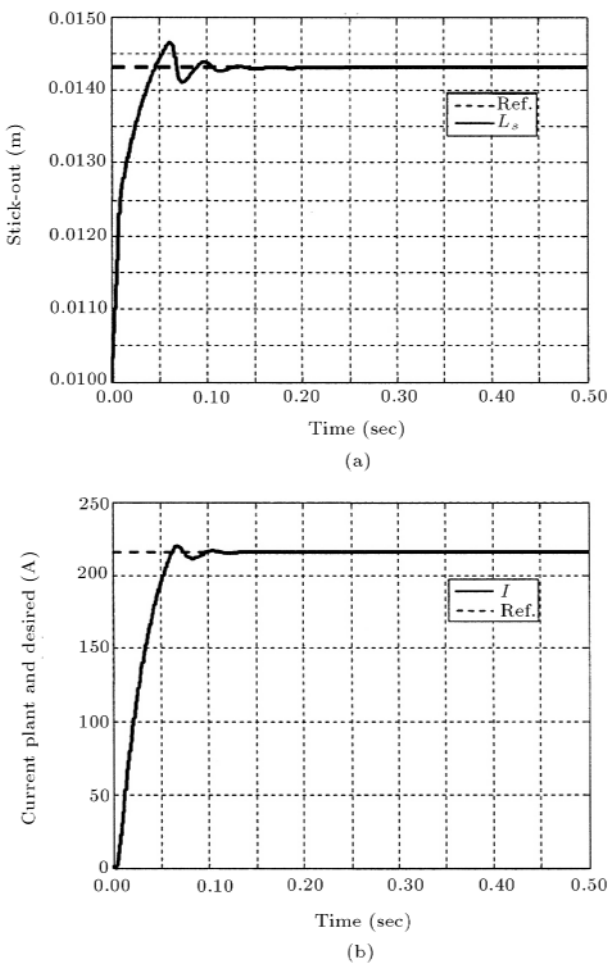


Figure 8. Desired and actual responses for stick-out (a) and current (b) for Case 1 of neural network controller with 50% model uncertainties.

weights are changing during the adaptation time until the structure of the network has become stable and can overcome the changes.

In Figure 10, the system and controller are simulated for cases where 70% model uncertainties, with external disturbance and noise, occurred in the welding process at time 0.5 sec. In this new case, only 4 parameters, E_a, ρ, R_a and L_s, of the welding process are examined as uncertainties. Indeed, external noise and disturbance, such as random or repeated signals, are applied to the control loop, which come from unpredictable effects in the control signal flow. It can be understood from Figures 8, 9 and 10 that the neural network controller has a robust behavior and an adaptation capability to overcome all unpredicted effects, due to model uncertainties, environmental disturbances and so on.

MIXED FUZZY CONTROL

The difficulty associated with applying a traditional fuzzy control theory for controlling MIMO systems, such as in many types of robotic manipulator, is in overcoming the coupling effects between the degrees of freedom.

Therefore, the concept of a coupling controller to compensate for these coupling effects was developed to enhance the control performance of robotic manipulators, such as the SCARA robot.

A typical dynamic model of a robotic system is complicated with uncertainty, so, model-free intelligent control strategies are employed in designing a MIMO system controller. This work proposes a new control approach by combining a TFC and a suitable coupling fuzzy controller for controlling MIMO systems. Adaptive control and robust control methods have been developed for controlling SISO and MIMO systems in order to evaluate their control performance [28,29]. However, the applications of these control methods still require some knowledge about the system and the variation range of the parameters; also, their output response has oscillatory features during the initial learning interval. The fuzzy set theory has been successfully applied in a number of control applications [30,31],

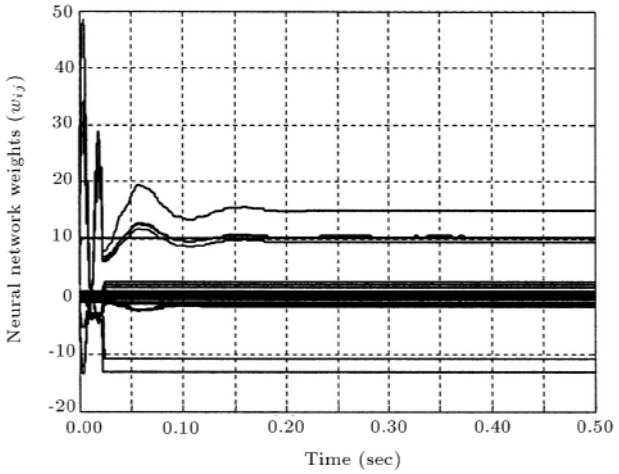


Figure 9. Changes in the weight of neural network controllers for Case 1 with 50% model uncertainty (the neural network becomes stable after 0.2 sec and overcomes the changes).

based on SISO systems, without considering the dynamics model of the system. This work develops a mixed fuzzy control strategy to control the robotic part of the system. The control strategy includes a TFC and a coupling fuzzy controller. Figure 11 shows this control block diagram. The procedure for designing the mixed control strategy involves, first, designing a TFC to individually control each degree of freedom of the robot. Then, an appropriate coupling fuzzy controller is designed to compensate for the system dynamic coupling between the degrees of freedom.

A TFC that operates with an output error of the system and an error differential in the continuous time system is adopted as the main controller to control each degree of freedom of the robotic system. Here, the input variable of the TFC, for the degrees of freedom of the system, are defined individually as follows:

$$e_i = R_i \quad Y_i, \tag{23}$$

$$\dot{e}_i = \dot{R}_i \quad \dot{Y}_i, \tag{24}$$

where e_i is the output state error of the ith degree of the system and $\dot{e}_i$ is used for indicating the state error derivative of the ith degree of a robot. The membership function of a fuzzy logic controller is usually difficult to decide. This work develops a membership function with an adjustable parameter, α_i^j, according to the dynamic characteristics of the selected robotic system. The control purpose can be achieved easily by only adjusting this parameter value, α_i^j, in the practical control process. Figure 12 represents a triangular membership function and its implementation to convert the input and output variables, e_i, $\dot{e}_i$ and u_i, into linguistic control variables (NB NM, NS, ZO, PS, PM, PB), where u_i is the control signal and α_i^j is

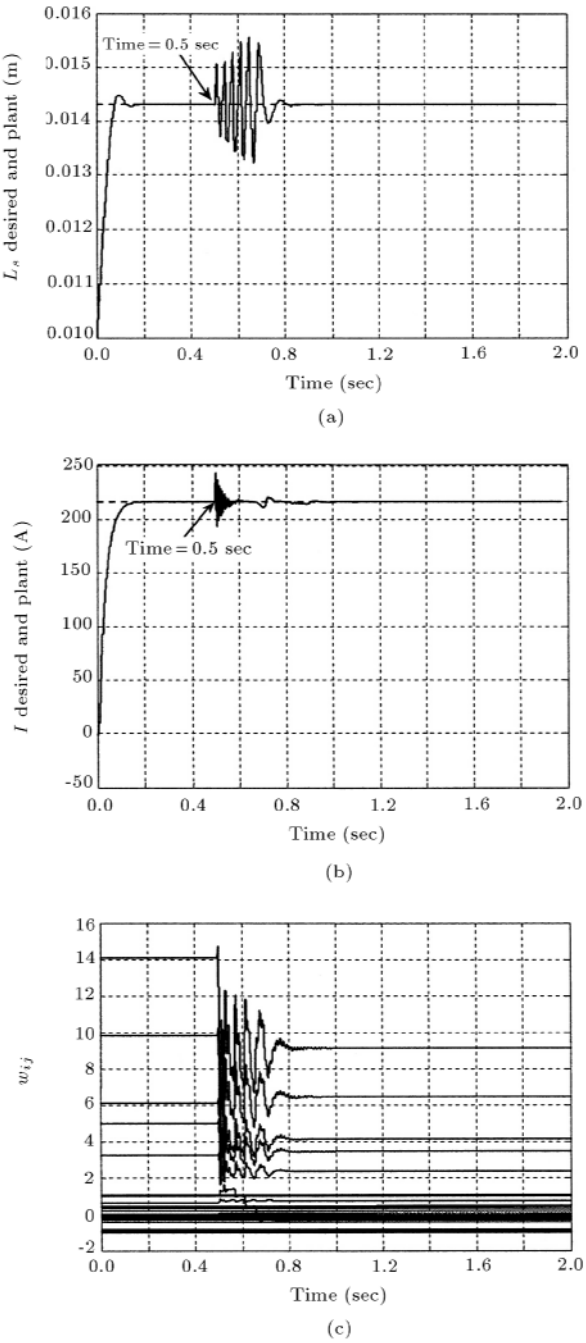


Figure 10. Desired and actual responses for stick-out (a), current (b) and neural network weights (c) for Case 1 with 70% uncertainty and noise occurring at 0.5 sec.

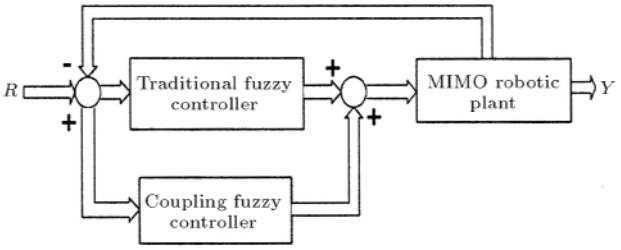


Figure 11. MFC for robotic system.

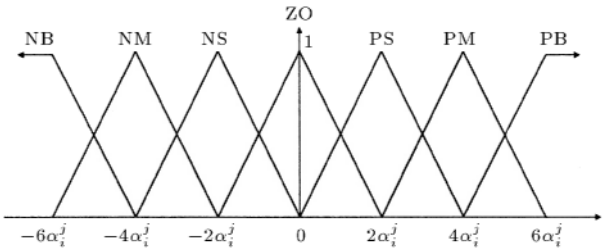


Figure 12. Membership function of a TFC.

a scaling factor. The subscripts $i = 1, 2, \cdots, m$ are used for expressing each degree of freedom of the system and the superscripts $j = 1, 2, 3$ are used for indicating the output state error of the system, the state error derivative and the control output. This study applies the state evaluation fuzzy control rule [30] for controlling robotic systems. Table 3 lists the fuzzy control rules that are employed in controlling a MIMO robotic system. The fuzzy inference logic applied the Max-Min product composition [30] to operate the fuzzy control rules. Finally, this work employs the "center of area" method [30], to defuzzify the output variables to achieve an actual control signal for the control task.

In a real MIMO system, the control output is influenced by more than one variable. According to the system characteristics, these system variables are obviously interactive as follows:

$$Y_i = P_i(u_1, u_2, \cdots, u_m, Y_1, Y_2, \cdots, Y_n), \qquad (25)$$

or:

$$Y_l = P_l(u_l, Y_1, Y_2, \cdots, Y_n), \qquad (26)$$

where $P_i(.)$ and $P_l(.)$ represent complex coupling functions that are difficult to define and drive. According to an analysis of the dynamics Equations 25 or 26, there exist u_i, the effect of which can be completed by u_l. u_i and u_l are the effects of the dynamics coupling characteristics of the MIMO system. The main effect on the system is controlled using a TFC. The secondary effect on the system is controlled by designing an appropriate coupling fuzzy controller. Clearly, the control structure of the fuzzy control system is very complicated, when the input variable is multiple-degree and the output variable is one, or more than one, degree. Thus, the parameters of a fuzzy control system are difficult to decide, because the fuzzy control rules are generated as a geometric series and, as a result, CPU time will increase significantly. Hence, the design procedure of this fuzzy coupling controller should be simplified to reduce the computational time during the implementation. The input variable of the coupling fuzzy controller is chosen, mainly, as two coupling affected factors, such that the structure of the fuzzy rules can be built conventionally.

According to the coupling situation between the degrees of freedom of the MIMO robotic systems, which can be analyzed by dynamics coupling characteristics, a few of the mainly affected coupling factors are, first, considered to design an appropriate coupling fuzzy controller in order to compensate for the dynamics coupling effects and enhance the control performance of the system. Figure 13 shows the membership functions of the coupling fuzzy controller, where β_i^j is a positive number scaling factor. According to the dynamics coupling characteristics of the robotic system, its meaning is the same as the scaling factor, α_i^j, in the previous section.

According to each degree of freedom of the robot and the effects of the dynamics coupling in the system, the fuzzy control rules of the coupling fuzzy controller can be decided and adjusted on the basis of the system's output response. Table 4 lists the coupling fuzzy control rules in controlling a MIMO robotic system. The fuzzy inference and defuzzification methods are similar to those of the TFC.

The outputs of the coupling fuzzy controller are chosen directly as the coupling control inputs. The main reason for this is that there is a different cou-

Table 3. Fuzzy control rules of a TFC.

	NB	**NM**	**NS**	**ZO**	**PS**	**PM**	**PB**
NB	NB	NB	NB	NM	NM	NS	ZO
NM	NB	NB	NM	NM	NS	ZO	PS
NS	NB	NM	NM	NS	ZO	PS	PM
ZO	NM	NM	NS	ZO	PS	PM	PM
PS	NM	NS	ZO	PS	PM	PM	PB
PM	NS	ZO	PS	PM	PM	PB	PB
PB	ZO	PS	PM	PM	PB	PB	PB

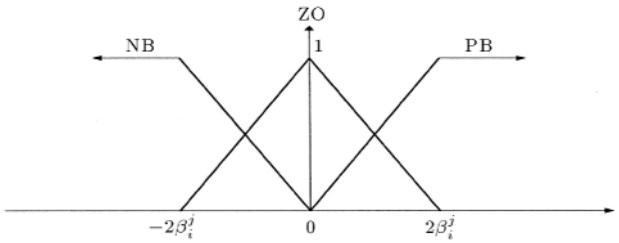


Figure 13. Membership function of a MFC.

Table 4. Fuzzy control rules of a MFC.

	NB	**ZO**	**PB**
NB	NB	NB	ZO
ZO	NB	ZO	PB
PB	ZO	PB	PB

pling effect for each step and it does not have an accumulating characteristic. The coupling effects are incorporated into the TFC for each step to improve the performance and robustness of the system. Accordingly, the total control input of this MFC is represented as follows:

$$U_i = u_i + U_{l \to i}, \qquad i \neq l, \tag{27}$$

where u_i expresses the system control input of the ith degree of a TFC and $U_{l \to i}$ represents the coupling effects control of the lth degree, relative to the ith degree of the coupling fuzzy controller.

The additional coupling fuzzy controller compensates for the unknown coupling effects of the MIMO robotic system, so, this study proposes a mixed fuzzy control strategy that can be employed to control MIMO robotic systems, such as SCARA. Huang and Lian [32] once proposed the mixed fuzzy control strategy and applied it in controlling the active dynamic absorber for an MIMO system and obtained good control performance. Lian and Lin [30] proposed a mixed fuzzy controller for general MIMO systems and, also, proposed a state-space approach for two DOF robotic cases to analyze the stability of the controller. In this work, previous ideas for mixed control strategies are used and, also, a GA is incorporated to design a more accurate and more performable mixed fuzzy controller, which is very suitable for controlling MIMO robotic plants and shows very interesting characteristics (which will be discussed in the next section). The dynamics equation of the robotic manipulator (Figure 14) is as follows:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau, \tag{28}$$

where $q = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T$ is a 3×1 vector of the joint position, $\dot{q} = \begin{bmatrix} \dot{q}_1 & \dot{q}_2 & \dot{q}_3 \end{bmatrix}^T$ is a 3×1 vector of the joint velocity, $\ddot{q} = \begin{bmatrix} \ddot{q}_1 & \ddot{q}_2 & \ddot{q}_3 \end{bmatrix}^T$ is a 3×1 vector of the joint acceleration, $\tau = \begin{bmatrix} \tau_1 & \tau_2 & \tau_3 \end{bmatrix}^T$ is a 3×1 vector of control input torque, $M(q)$ is a 3×3 inertial matrix, $C(q,\dot{q})$ is a 3×3 matrix of coriolis and centrifugal forces and $G(q)$ is a 3×1 gravity vector.

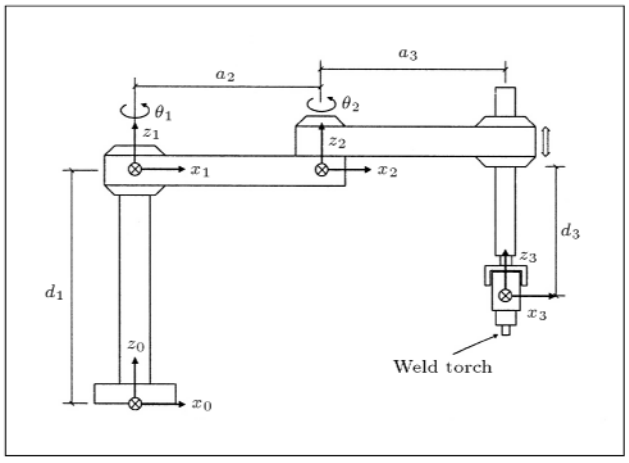


Figure 14. Schematic diagram of a three-axis SCARA robot.

In this paper, it is assumed that the three links of the robotic manipulator are thin cylinders or rods of mass, m_1, m_2 and m_3, respectively. In general, due to the robust behavior of the designed controller, if the dynamic equation of the robot is expressed as follows:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \text{Unknown} = \tau, \tag{29}$$

the controller is still capable of overcoming the undesired effects of unknown terms and disturbances, which may come from joint friction and environmental vibrations etc. Because of the differences in the scope of this work and these effects, it is not necessary to consider uncertainties and disturbances in the simulation of the robot. Thus, this work only studies the tracking response of the robotic manipulator and the optimization process for this task. The kinematics Denavit-Hartenberg parameters of the SCARA robot are given in Table 5 [33].

Using the inward and outward Newton-Euler iterational method, proposed in [33], the joint torques will be derived in a close form as follows:

$$\begin{aligned} \tau_1 = {} & \left[\left(\frac{m_{+1}}{3} + m_2 + m_3 \right) a_2^2 + (m_2 + 2m_3) a_2 a_3 C_2 \right. \\ & \left. + \left(\frac{m_2}{3} + m_3 \right) a_3^2 \right] \ddot{\theta}_1 \quad \left[\left(\frac{m_2}{2} + m_3 \right) a_2 a_3 C_2 \right. \\ & \left. + \left(\frac{m_2}{3} + m_3 \right) a_3^2 \right] \ddot{\theta}_2 + b_1(\dot{\theta}_1) \\ & a_2 a_3 S_2 \left[(m_2 + 2m_3)\dot{\theta}_1 \dot{\theta}_2 \quad \left(\frac{m_2}{2} + m_3 \right) \dot{\theta}_2^2 \right]. \end{aligned} \tag{30}$$

The torque on the second joint is given by:

$$\begin{aligned} \tau_2 = {} & \left[\left(\frac{m_2}{2} + m_3 \right) a_2 a_3 C_2 + \left(\frac{m_2}{3} + m_3 \right) a_3^2 \right] \ddot{\theta}_1 + b_2(\dot{\theta}_1) \\ & + \left(\frac{m_2}{2} + m_3 \right) a_3^2 \ddot{\theta}_2 \quad a_2 a_3 S_2 \left(\frac{m_2}{2} + m_3 \right) \dot{\theta}_1^2. \end{aligned} \tag{31}$$

Table 5. Kinematics parameters of the three-axis SCARA robot.

Joint	d	a	θ	α	Home
1	d_1	0	θ_1	0	0
2	0	a_2	θ_2	0	0
3	d_3	a_3	0	0	0

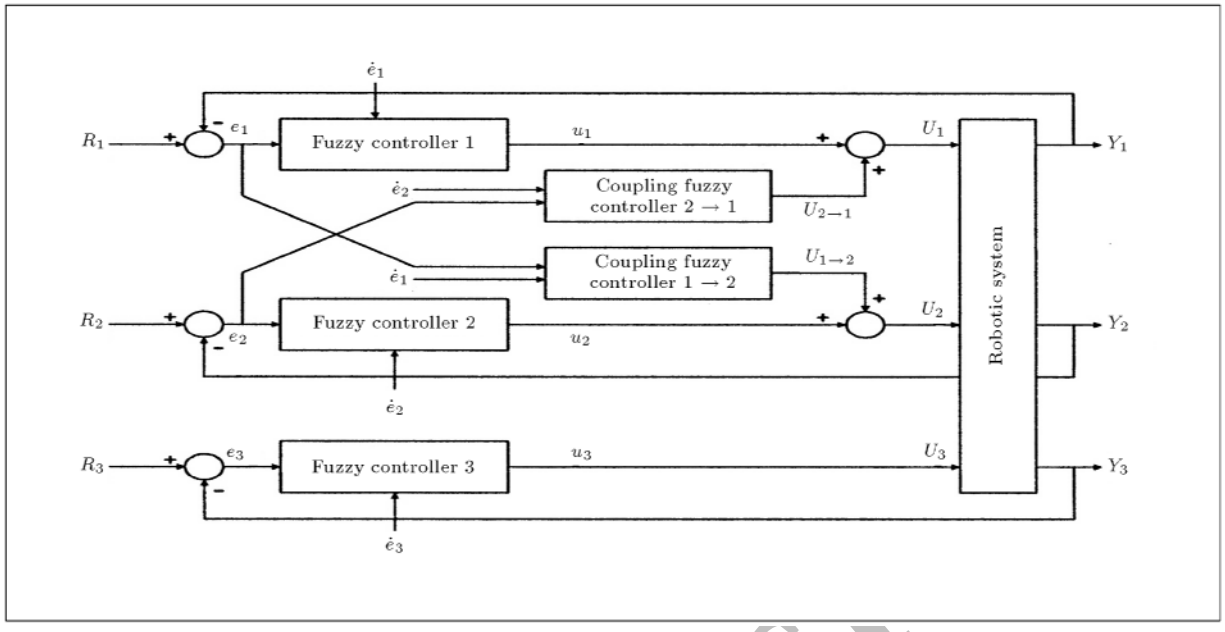


Figure 15. Block diagram of the mixed and traditional fuzzy control strategy.

And the torque on the third joint is given by:

$$\tau_3 = m_3\ddot{d}_3 \quad gm_3 + b_3(\dot{d}_3), \tag{32}$$

where (g) represents gravity acceleration and b_1, b_2 and b_3 represent the robotic manipulator, based on a typical welding task, are chosen as:

$$m_1 = 2 \text{ kg}, \quad m_2 = 3 \text{ kg}, \quad m_3 = 3 \text{ kg},$$
$$a_2 = 0.4 \text{ m}, \quad a_3 = 0.6 \text{ m}, \quad d_3 = 0.5 \text{ m},$$

The three-axis SCARA robot is treated as a system, comprised of a two-input two-output system (for the first two links) and a single-input single-output system (for the last link). The dynamics of Equations 30 and 31 are coupled and the first two joints are considered to be a MIMO system. Hence, a mixed fuzzy controller (Tables 3 and 4) has to be designed to control the first two joints of the robot.

The third joint is dynamically de-coupled from the first two joints and is modeled as a SISO system. Hence, a traditional SISO fuzzy controller (Table 3) has to be designed to control the third joint of the robot. Figure 15 represents the control block diagram of the designed fuzzy controller. The angular errors and error derivatives of the links 1, 2 and 3, defined by Equations 23 and 24, are chosen as the input variables of a TFC. From Equation 27, the total control input torques for this controller are given by:

$$U_1 = u_1 + U_{2\to1}, \qquad U_2 = u_2 + U_{1\to2}, \qquad U_3 = u_3, \tag{33}$$

where u_1, u_2 and u_3 are the control input torques that come from the TFC of the links 1, 2 and 3, respectively. $U_{1\to2}$ and $U_{2\to1}$ represent the torques, due to the coupling effects between links 1 and 2, respectively.

In this work, two sets of desired trajectories for the simulation of a SCARA robot are used. The first set is for examining the performance of the designed controller, along a sinusoidal path, $R_1 = p_1 \sin(2\pi t/T_d)$ and $R_2 = p_2 \sin(2\pi t/T_d)$, which is shown in Figures 16 through 19, and the second set is for the weld profile, which is discussed in the next section. For the first

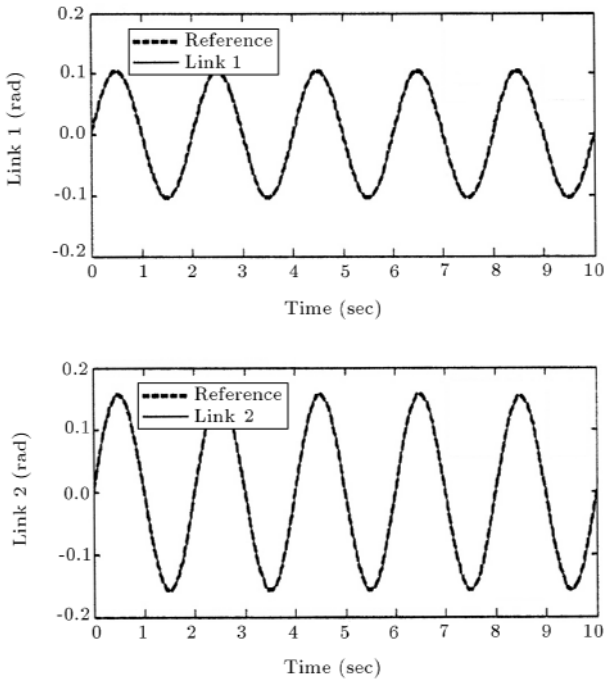


Figure 16. Angular tracking response using a TFC.

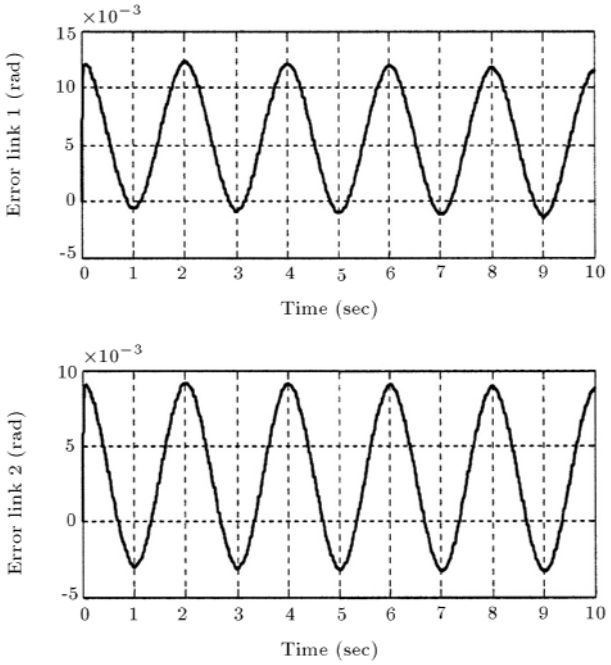


Figure 17. Angular tracking error response using a TFC.

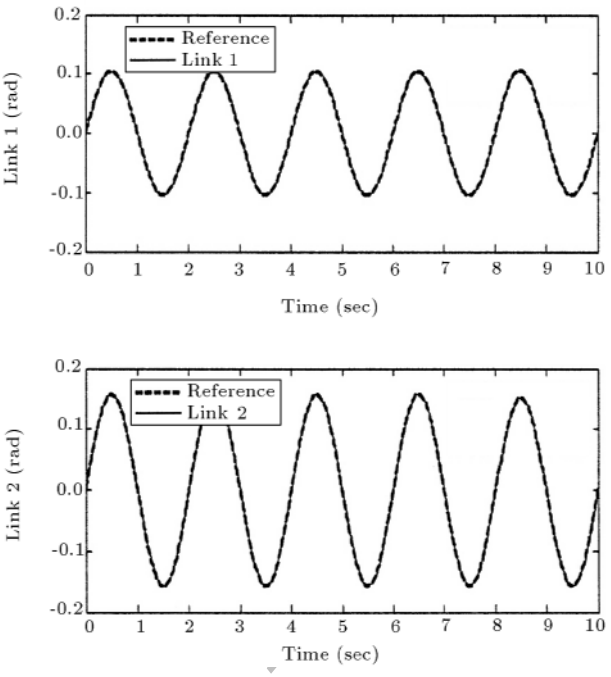


Figure 18. Angular tracking response using a MFC.

case, the period, T_d, is 2 sec and the amplitudes are chosen as $p_1 = 6°$for the first link and $p_2 = 9°$for the second link. Parameters of the TFC were chosen as follows:

$$\alpha_i^j = \begin{cases} \alpha_1^1 = 1.2, \quad \alpha_1^2 = 0.02, \quad \alpha_1^3 = 3, \\ \alpha_2^1 = 1.2, \quad \alpha_2^2 = 0.02, \quad \alpha_2^3 = 1.5, \end{cases}$$

Parameters of an appropriate coupling fuzzy controller

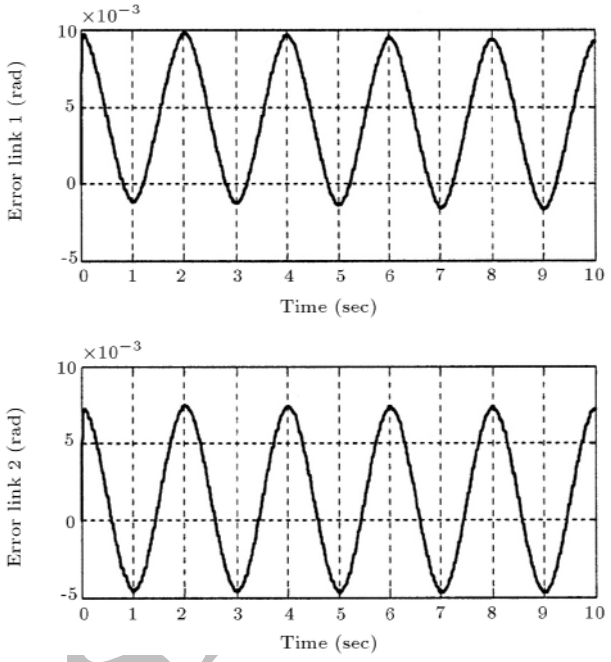


Figure 19. Angular tracking error response using a MFC.

were chosen as follows:

$$\beta_i^j = \begin{cases} \beta_1^1 = 1.1, \quad \beta_1^2 = 0.02, \quad \beta_1^3 = 3, \\ \beta_2^1 = 1.1, \quad \beta_2^2 = 0.04, \quad \beta_2^3 = 1.5. \end{cases}$$

Figures 16 and 18 plot the angular tracking response of the TFC and MFC. According to the angular tracking errors of these two controllers, which are shown in Figures 17 and 19, the control performance of these controllers can be compared further. The maximum amplitude of the angular tracking error is within 0.0097 rad for link 1 and 0.0065 rad for link 2 of this robotic system for MFC, which is compared to 0.012 rad and 0.009 rad for the first and second links, respectively, of the TFC. Clearly, the control performance of the mixed fuzzy control algorithm exceeds that of the TFC for this MIMO robotic system.

INTEGRATION OF MIXED FUZZY LOGIC CONTROLLER AND GENETIC ALGORITHM

A Genetic Algorithm (GA) is an optimization technique developed by J. Holland et al. A GA is one of the recently attracted search algorithms, based on the mechanics of natural selection and natural genetics [34]. It is easy to implement GA as an efficient mothod for multivariable optimization problems, such as fuzzy controller design. A GA uses the operations of selection, reproduction, crossover and mutation to generate the next population and it searches for an optimal solution from a set of points. Recently, many

GAs have been presented. Figure 20 shows a flow chart of the GA.

The contents of a conventional GA can be simply described as follows:

Coding of Parameters: The strings of searching parameters for the optimization problem are the genes in a chromosome, which can be binary-coded or real coded;

Initial Generation: Generates an initial population of N strings by calling random numbers;

Fitness Evaluation: The fitness function is the performance index of a GA for verifying the solutions;

Selection: The fitness of offspring is compared with that of the parent. The better chromosome is selected and the worst is killed;

Reproduction: Is a copy of the strings from a mating pool into a new generation. Strings with high fitness values obtain a larger number of copies in the next generation;

Crossover: Selecting random pairs of strings from a mating pool and some portions of the chromosomes are exchanged;

Mutation: A part of a chromosome is changed and makes it probable that the population will get out of the local minimum.

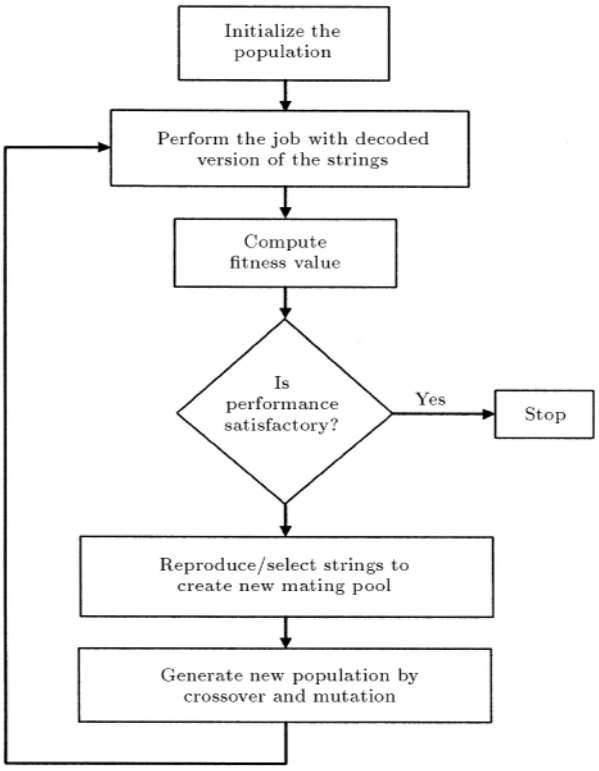


Figure 20. Flow chart of the genetic algorithm [20].

The GA is a powerful tool for structure optimization of the fuzzy logic controller. In this paper, the GA is used for optimization of the mixed fuzzy controller, which is undertaken in two steps. In step 1, the GA is used to find an optimal membership function and, in step 2, the torque ranges and scaling factors of the first and second links of the robot in the MFC are optimized. The membership functions can be represented either by triangular or trapezoidal shapes. In this study, triangular fuzzy membership functions are used. These triangles include the variable based width and shift along the x-axis, freely. Therefore, each one requires the definition of two points to fix it. There are three ways to achieve optimal membership functions:

1. Choose all membership functions as variables to be optimized,
2. Choose only the overlaps between membership functions as variables to be optimized,
3. Choose only the position of the bottom vertices of each membership triangle and do not change the tip position.

In the first choice, the entire set of fuzzy membership functions for a MFC system must be represented as bit strings (of 0 and 1). This can be a lengthy and complicated procedure. To do this, the best choice is to use a method called concatenated, mapped, unsigned, binary coding [34]. In the second choice, the overlaps between different fuzzy membership functions are considered as parameters to be optimized. Unlike the first choice, all the fuzzy membership functions are not needed to be coded into bit strings [18]. In the third choice, two bottom vertices of each triangle are considered as variables. This method is very efficient, if the membership functions are symmetric. In this case, the number of variables to be changed has become half and this makes utilization of the GA for fuzzy membership optimization much easier and more efficient. In this paper, the third method for the optimization process is used.

In order to define the membership functions, variables must be considered, such that the formulation of the membership functions becomes dependent on those variables. For this reason, here, the method in Figure 21 is considered. As shown in the figure, each triangle is formulated, based on the position of its tip and parameters, a_i^j, b_i^j and c_i^j. In these parameters, index i is dedicated to the usage of those membership functions, as for the first or second link controller; index j is for the sequence of triangles in the specific membership function and parameters a, b and c are for error, derivative error and torque, respectively. The only constraint imposed on the individual triangle of the membership functions is that the triangles bordering the extreme limits of the action or control

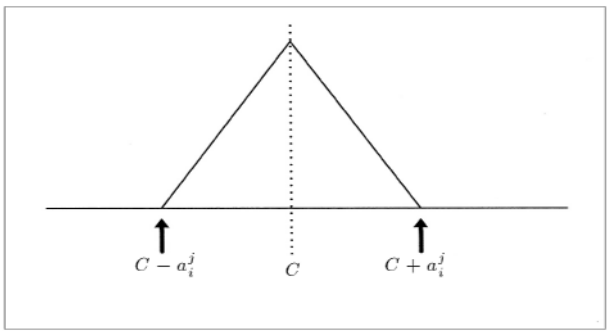


Figure 21. Defining variables for triangles of membership functions.

must not be changed. This is because, for almost all applications of FLC systems, the membership functions have two extreme limits (i.e., upper bound and lower bound), which are shown in Figure 22.

The extreme bordering triangles are called corner triangles and the triangles between the two extreme bordering triangles are called inner triangles. Each corner triangle needs the definition of only one point for it to be determined, while each inner triangle needs the definition of two points for it to be determined. For optimization of the membership functions of a FLC, it is only required to code the variables, a_i^j, b_i^j and c_i^j, into bit strings. Here, in this paper, only the membership functions of the first and second joint torques in the TFC section of the mixed fuzzy controller are optimized, thus, there are 12 variables to construct the bit strings. These 12 variables are $c_1^{1\to 6}$ and $c_2^{1\to 6}$. Note that variables a and b are not used.

The bit strings representing these parameters must then be judged and assigned a fitness value, which is a score representing the degree to which they accomplished the goal of defining high performance. In this paper, the Root Mean Square Error (RMSE) for the first and second links of the robot (Equation 34) is used as the fitness function of the genetic algorithm. The RMSE term can be evaluated to determine the fitness of the strings in a population. This error could be the distance between the set point and the output of the system. Here, 30 populations and 100 generations are used for the optimization process.

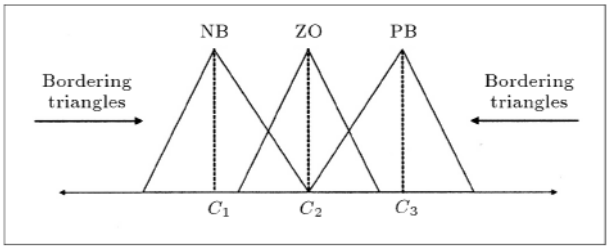


Figure 22. A set of membership functions of a FLC with bordering triangles of the membership functions.

In step 2 of the optimization process in this paper, the torque ranges and scaling factors of the first and second links in the MFC are optimized. In this part of the optimization process, eight variables, formerly identified as α_1^3, α_2^3, β_1^3 and β_2^3, and ranges for u_1, u_2 and $U_{1\to 2}, U_{2\to 1}$, in designing membership functions, are considered. The bit strings representing these parameters must then be judged and assigned a fitness value. The results of the optimization processes for both steps 1 and 2 are shown in Figures 23 through 25 and the fitness function is as follows:

$$\text{Fitness} = \left(\left(\sum_{k=1}^{n_1} \text{error}_{1k}^2 \right) / n_1 + \left(\sum_{k=1}^{n_2} \text{error}_{2k}^2 \right) / n_2 \right)^{\frac{1}{2}}, \quad (34)$$

where n_1 and n_2 represent the size of the error vector for links 1 and 2, respectively, which are equal here, and error_{1k} and error_{2k} are first and second link errors. The values of optimized scaling factors are as follows:

$$\alpha_1^3 = 3.66, \qquad \alpha_2^3 = 3.314,$$

and

$$\beta_1^3 = 3.91, \qquad \beta_2^3 = 1.39.$$

According to Figure 24, the maximum amplitude of the angular tracking errors are within 0.004589 rad for link 1 and 0.0019 rad for link 2, which can be compared to 0.0097 rad for link 1 and 0.0074 rad for link 2 before optimization (Figure 19). Clearly, the control performance of the mixed fuzzy GA control algorithm grows, rather than the MFC.

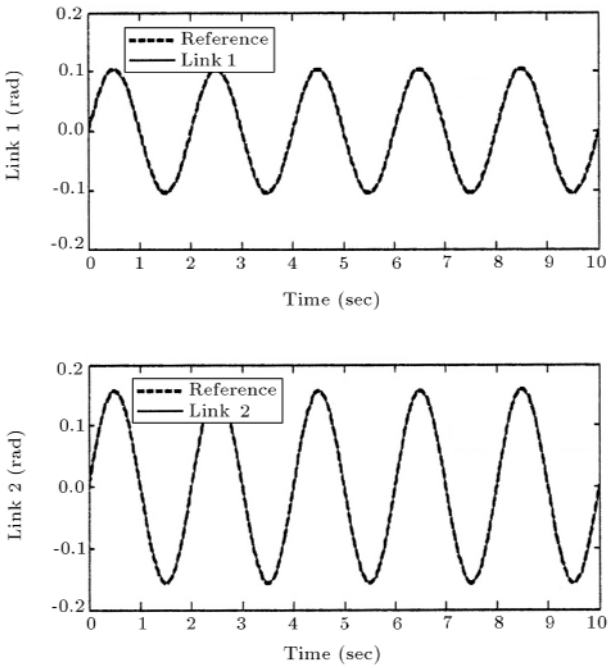


Figure 23. Angular tracking response using an optimized MFC.

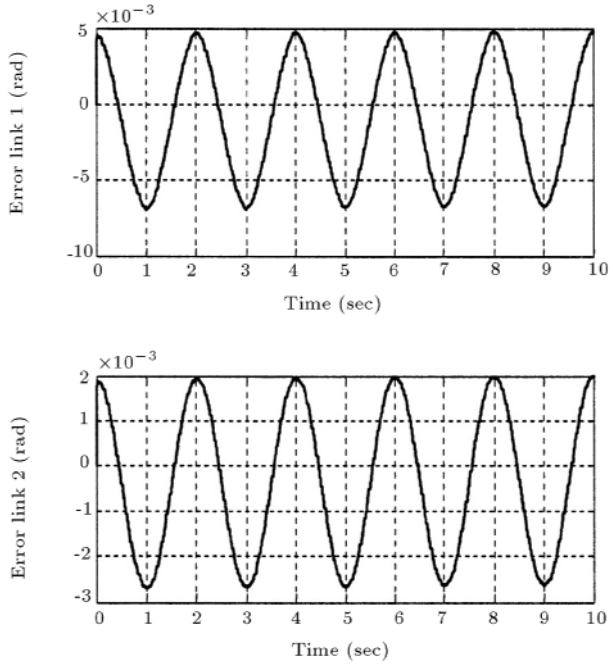


Figure 24. Angular tracking error response using an optimized MFC.

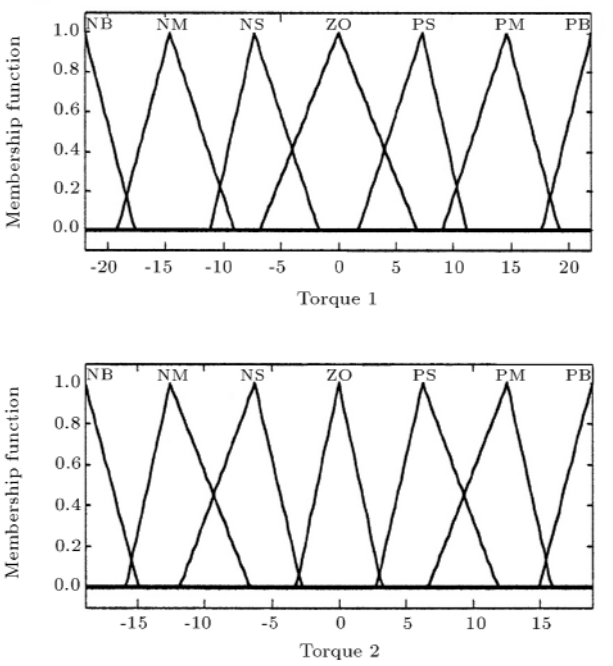


Figure 25. Modified membership functions of an optimized MFC for links 1 and 2 in TFC part of the controller.

SIMULATION RESULTS OF THE WELDING ROBOT

The inference between the GMAW and the SCARA, as discussed in the earlier part of the paper by Equations 9 through 11, obtains the value of the robot end-effector speed or the weld torch speed as input and computes the value of the desired stick-out, or arc voltage and desired current as outputs. For a simulation of the welding robot, Case 1 in Table 2 is considered here. The actual welding starts when the robot has reached the lift off position and goes on until the weld speed or the end-effector speed is constant at 8 mm/s.

According to the trajectory planning for the weld profile, the speed is constant from time $t = 0.1$ sec to $t = 2.5$ sec, but the third link reaches its desired trajectory after 0.25 sec. Hence, the robot executes the welding operation of the desired weld profile from $t = 0.25$ sec to $t = 2.5$ sec, for a period of 2.25 sec., at a speed of 8 mm/s. At $t = 2.5$ sec, the power is turned off and the welding ends. Hence, the desired stick-out and current responses are obtained. For obtaining the desired angular trajectory of the robotic system, the inverse kinematics transformation of the output position into the joint coordinate must be formulated. The relation between the position of the arm end-effector and the values of the joint angles are as follows:

$$x_e = a_2\cos(\theta_1) + a_3\cos(\theta_1+\theta_2),$$
$$y_e = a_2\sin(\theta_1) + a_3\sin(\theta_1+\theta_2), \tag{35}$$

where x_e and y_e represent the end-effector position. An inverse mapping of linkage coordinates to joint coordinates must be performed, so an inverse problem has to be solved by using Equation 35. Joint angles θ_1 and θ_2 can be expressed in terms of the tip position as follows:

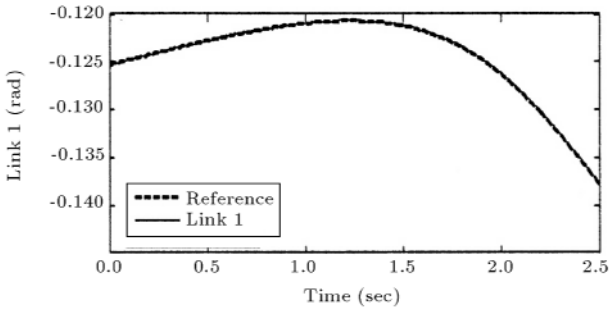
$$\cos(\theta_2) = \frac{(x_e^2+y_e^2) \quad (a_2^2+a_3^2)}{2a_2a_3}, \tag{36}$$

$$\tan\theta_1 = \frac{(a_3\sin\theta_2)x_2 + (a_2+a_3\cos\theta_2)y_e}{(a_3\sin\theta_2)y_e + (a_2+a_3\cos\theta_2)x_e}. \tag{37}$$

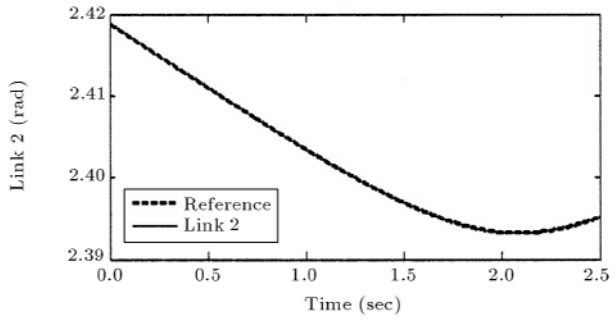
The results of the weld profile simulation are shown in the following figures. For tracking the weld profile, the speed of the weld torch is holding constant at 8 mm/s and the duration time of the welding is chosen as 2.5 sec. Note that the desired weld profile is curvilinear. Figure 26 shows the joint trajectories and Figure 27 shows the tracking response of the GMAW system.

Once the desired response is achieved for the angular positions of the first and second joints, it is important to have the third joint in such a position that the proper weld torch position is available for welding. The third joint moves from 0.2 m to 0.49 m and Figure 26c shows that the desired response is obtained for the third joint.

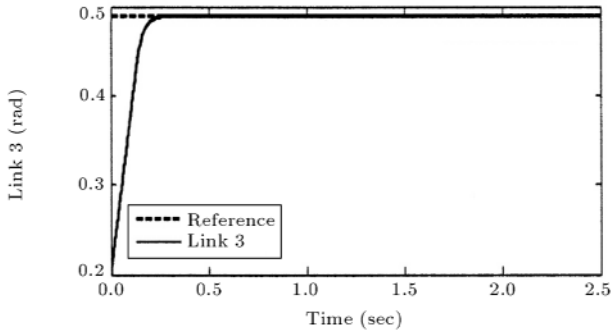
For validation of this work in comparison to previous work in this area, some simulations are presented here. The first simulation is a comparison between ANNC and the feedback linearization controller [3] of the GMAW process, in cases where 50% model



(a) Tracking response of the first link



(b) Tracking response of the second link



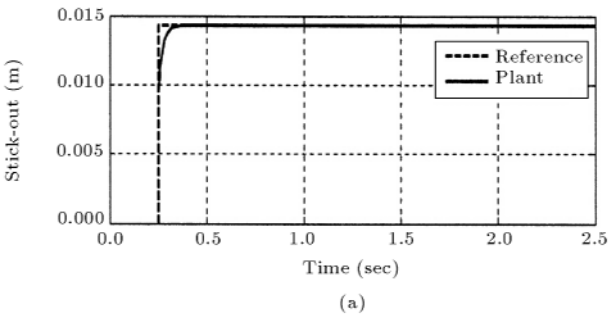
(c) Tracking response of the third link

Figure 26. Tracking responses of robot joints for the weld profile.

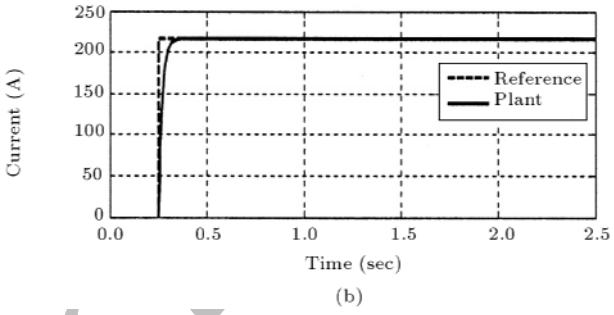
uncertainty exists in the plant parameters. Figure 8, formerly discussed, shows that tracking is accomplished successfully after almost 0.2 sec. But, Figure 28, which is based only on the feedback linearization strategy discussed in [3], cannot converge to the desired trajectory, even after 0.5 sec, and a steady state error exists in the system response. Another simulation is presented in Figures 18 and 23, which shows a difference between the TFC [13,14,30] and MFGAC. From these figures, it is also clear that the tracking response of the MFGAC is much better than that of the TFC.

CONCLUSION

In this paper, the modeling of a SCARA type robot for a continuous welding job was addressed. Ad-

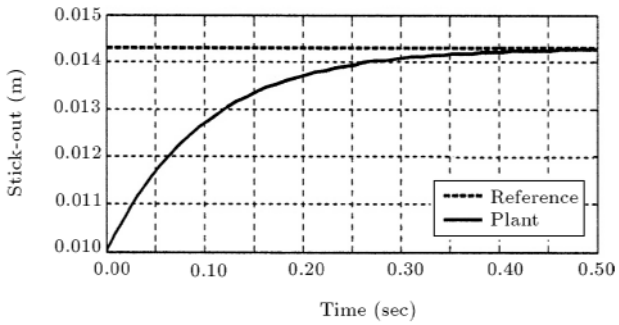


(a)



(b)

Figure 27. Stick-out (a) and current (b) responses of the system for Case 1 in Table 2 of the weld profile.



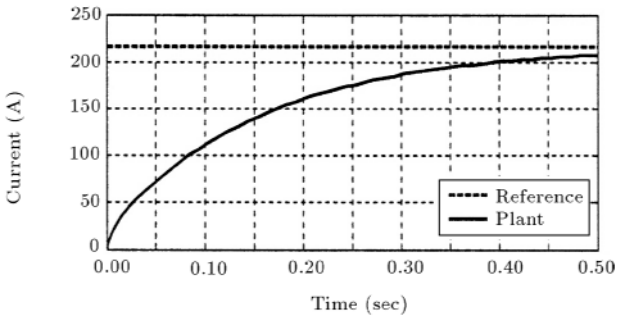


Figure 28. Desired and actual responses for stick-out (a) and current (b) for Case 1 in Table 2 of feedback linearization controller with 50% model uncertainties.

vanced control strategies, including feedback linearization mixed with a neural network and fuzzy controller and mixed with a GA, have been used to accomplish control tasks for both robot motion and a welding job, together. From the implemented control algorithm based adaptive neural network, it can be seen that

very good performance in the weld task is achieved. In the control strategy of the robotic manipulator, the mixed fuzzy control and GA perform very well and the controller's performance and robustness are improved, compared to a simple fuzzy controller.

ACKNOWLEDGMENT

The authors gratefully acknowledge the assistance of the Department of Mechanical Engineering at Sharif University of Technology.

REFERENCES

1. Ozcelik, S., Moore, K.L. and Naidu, S.D. "Application of MIMO direct adaptive control to gas metal arc welding", *Proceeding of the American Control Conference*, Philadelphia, USA (June 1998).
2. Smartt, H.B. and Einerson, C.J. "A model for heat and mass input control in gas metal arc welding", *Welding Journal*, pp 217-229 (May 1993).
3. Abdel Rahman, M. "Feedback linearization control of current and arc length in GMAW systems", *Proceeding of American Control Conference*, Philadelphia, USA (June 1998).
4. Jalili-Kharaajo, M., Gholampour, V., Ebrahimirad, H. and Esna Ashari, A.R. "Robust nonlinear control of current and arc length in GMAW systems", *Proceedings of IEEE Conference on Control Applications* (June 2003).
5. Henderson, D.E., Kokotovic, P.V., Schiano, J.L. and Rhode, D.S. "Adaptive control of an arc welding process", *IEEE Control Systems Magazine*, pp 49-53 (1993).
6. Doumanidis, C.C. "Multiplexed and distributed control of automated welding", *IEEE Control Systems*, pp 13-24 (1994).
7. Cook, G.C., Anderson, K. and Barrett, R.J. "Feedback and adaptive control in welding", In *2nd Intl. Conf. on Trends in Welding Research*, S.A. David and J.M. Vitek, Eds., pp 891-903, Gatlinburg, TN, USA (1989).
8. Moore, K.L., Yender, R., Tyler, J. and Naidu, D.S. "Modeling, calibration, and control-theoretic analysis of the GMAW process", *Proceeding of American Control Conference*, Philadelphia, Pennsylvania, USA (June 1998).
9. Golob, M., Koves, A., Puklavec, A. and Tovornik, B. "Modeling, simulation and fuzzy control of GMAW welding process", In *15th Triennial World Cong. Int. Fed. of Automatic Control* (2002).
10. Bingul, A. and Cook, G.E. "Dynamic modeling of GMAW process", *Proceedings of the 1999 IEEE International Conference on Robotics & Automation*, Detroit, Michigan, USA (May 1999).
11. Neo, N.N. and Er, M.J. "Adaptive fuzzy controllers of robot manipulator", *Int. J. Syst. Sci.*, **27**(6), pp 519-532 (1997).
12. Wang, S. and Tsuchiya, T. "Fuzzy trajectory planning and its application to robot manipulator path control", *Adv. Robot.*, **8**(1), pp 73-93 (1994).
13. Tang, W. and Chen, G. "Robust fuzzy controller for a flexible-joint robot arm with uncertainties", in *Proceedings of the IEEE International Conference on Fuzzy Systems*, Orlando, FL, USA, pp 1554-1559 (1994).
14. Kim, E. "Output feedback tracking control of robot manipulators with model uncertainty via adaptive fuzzy logic", *IEEE Trans. Fuzzy Syst.*, **12**(3), pp 368-378 (2004).
15. Emara, H. and Elshafei, A.L. "Robust robot control enhanced by a hierarchical adaptive fuzzy algorithm", *Eng. Appl. Artificial Intell.*, **17**(2), pp 187-98 (2004).
16. Karr, C.L. and Gentry, E.J. "Fuzzy control of PH using genetic algorithms", *IEEE Trans. Fuzzy Syst.*, **1**(1), pp 46-53 (Feb. 1993).
17. Karr, C.L. "Design of an adaptive fuzzy logic controller using genetic algorithm", in *Proc. The Fourth Int. Conf. Genetic Algorithms*, pp 450-457 (1991).
18. Mohammadian, M. and Stonier, R.J. "Tuning and optimization of membership functions of fuzzy logic controllers by genetic algorithms", *IEEE International Workshop on Robot and Human Communication* (1994).
19. Mester, G. "Neuro-fuzzy-genetic controller design for robot manipulators", *Proceedings of the 1995 IEEE IECON 21st International Conference on Industrial Electronics, Control, and Instrumentation* (1995).
20. Taner Eskil, M., Onder Efe, M. and Kaynak, O. "T-norm adaptation in fuzzy logic systems using genetic algorithms", *Proceedings of the IEEE International Symposium on Industrial Electronics* (1999).
21. Moore, K.L., Naidu, D.S., Abdelrahman, M.A. and Yesildirek, A. "Advance welding control project", *Annual Report FY96. Technical Report*, ISU, Pocatello, ID, USA (1996).
22. Smartt, H.B., Johnson, J.J., Carlson, N.M. and Waddoups, M. "Dynamics of droplet detachment in GMAW", *3rd Intl. Conf. on Trends in Welding Research*, Gatlindurg, TN, USA (1992).
23. Campa, G., Fravolini, M.L., Napolitano, M. and Seanor, B. "Neural networks-based sensor validation for flight control system of a B777 research model", *Proceeding of the American Control Conference*, Anchorage, AK, USA, May 8-10 (2002).
24. Napolitano, M.R., Chen, C.I. and Naylor, S. "Aircraft failure detection and identification using neural networks", *AIAA Journal of Guidance control and Dynamics*, **16**(6), pp 999-1009 (1993).
25. Campa, G., Fravolini, M.L. and Napolitano, M. "A Library of adaptive neural networks for control purposes", *IEEE Int. Symposium on Computer Aided Control System Design*, September 18-20, Glasgow, Scotland, UK (2002).

26. Neural Network Toolbox user guide, *The Mathwork Inc. 1994-2006*, Available: http://www.mathworks.com/matlabcentral/fileexchange/(Oct. 2002)
27. *Mathwork Inc. Available*: http://www.mathworks.com/.
28. Chantranuwathana, S. and Peng, H. "Modular adaptive robust control of SISO nonlinear systems in a semi strict feedback form", *Int. J. Robust Nonlinear Control*, **14**(6), pp 581-601 (2004).
29. Laib, A. "Adaptive output regulation of robot manipulators under actuator constraints", *IEEE Trans. Robot Automat*, **16**(1), pp 29-35 (2000).
30. Lian, R.J. and Lin, B.F. "Design of a mixed fuzzy controller for multiple-input multiple-output systems", *Mechatronics*, pp 1225-1252 (2005).
31. Arisariyawong, S. and Charoenseang, S. "Reducing steady-state errors of a direct drive robot using neuro-fuzzy control", *Second Asian Symposium on Industrial Automation and Robotics* (May 2001).
32. Huang, S.J. and Lian, R.J. "Active vibration control of a dynamic absorber using fuzzy algorithms", *Mechatronics*, **6**(3), pp 317-336 (1996).
33. Craig, J.J., *Introduction to Robotics: Mechanics and Control*, Second Edition, Addison Wesley (1989).
34. Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, (1989).